

## TURNTABLE DRIVE - SM9468

---

A special version of the PM300 has been designed to accommodate the particular needs of a servo controller used in a turntable speed control application.

A switch connected to the PM300's *Fast Jog* input selects either manual mode where the turntable speed is set on thumbwheel switches, or computer mode where all the standard PM300 commands are available plus some application specific commands. The PM300 is set to axis address 0, therefore no command address prefix is required.

### **Special Commands:**

**QSCL** query current scale factor

**SCAL** set scale factor  
range: 4940 to 5461  
initial value: 5208

The maximum boundary of speed in steps/sec is 65535. This therefore sets the maximum speed x scale factor.  
i.e.  $5461 \times 1200 \div 100 = 65532$  steps/sec.

**RPM** set speed in r.p.m. x 10  
range: 100 to 1200 i.e. 10.0 to 120.0 r.p.m.

**RT** Read value on thumbwheel switch inputs

### **Computer Mode**

In computer mode the speed would first be set using the **RPM** command. The motor is then set in constant velocity mode using the **CV** command. To set the motor running in the opposite direction a **CV-1** command would be sent. To stop the motor once running a **ST** command would be used. The speed may be changed while in **CV** mode.

The scale factor that converts r.p.m. at the turntable into steps per second at the motor is set using the **SCAL** command. This may be changed while the motor is running to allow calibration.

An extra function added to the ERROR led on the front panel is *At Speed*. This led will illuminate while the motor is accelerating or decelerating to a new speed. The ERROR output is also accessible via the board connector.

### **Setting acceleration & deceleration**

The acceleration is set using the **SA** command and the deceleration is set using the **SD** command. Both values are set in steps/second<sup>2</sup> at the motor.

### **Servo coefficients**

The initial value of the *velocity coefficient* **KV** has been set to 80 and the initial value of the *proportional coefficient* **KP** has been set to 1500.

### **Manual Mode**

In manual mode the PM300 is also operating in CV mode but with the speed set by thumbwheel switches.

The run switch connected to the PM300's - *Jog* input sets the motor running. As in computer mode the front panel LED will go out when the motor is at speed. A Direction switch connected to the PM300's + *Jog* input may be used to set the direction of rotation before the run switch is activated.

### **Thumbwheel switch wiring**

The type of thumbwheel switches used are diode multiplexed BCD. The common connection to each switch is connected to one of the PM300's *Write Ports* and the BCD outputs are connected to the PM300's *Read Port* inputs.

The thumbwheel switch inputs may be tested using the **RT** command. This returns a numeric value corresponding to the thumbwheel switch settings.

PM300      PROGRAMMERS REFERENCE

CONTROL C (ASCII 03)      **Hard Stop.**

Moves, sequences and profiles halted immediately.  
Command buffer cleared.  
Error signal cleared.  
Sets status to **Idle**.  
Auto-execute flag cleared.  
Sequences and profiles retained in memory.  
Operates on all axes.

ESC (ASCII 27)      **Soft Stop**

Command buffer cleared.  
Motion stopped at the **SD** rate.  
Status returns to **Idle**.  
Auto-execute flag cleared.  
Sequences and profiles retained in memory.  
Operates on all axes.

<b>AA</b>	<b>Allow Abort</b>
-----------	--------------------

Allow abort mode. If the error between **CP** and **AP** exceeds the value of **TR** (tracking window) the controller latches the error signal, sets the analogue output signal to the amplifier to 0V, turns the Error LED on and aborts a move. The PM300 remains aborted until reset by sending **RS** or powering-down. The controller will in the meantime respond to queries only. If a move command is attempted the controller will respond with an ! TRACKING ABORT error message. This mode is the default if the controller is re-initialised.

In this mode, selected values for the tracking window will have to take into account the normal position error or *lag* that occurs during rapid acceleration.

<b>Syntax</b>	<b>Units</b>	<b>Range</b>	<b>to</b>	<b>Initial State</b>
<ad>AA	N/A	N/A		Enabled

<b>Condition Requirements</b>	<b>Notes:</b>
None.	Value retained on power-up.

<b>Response:</b>	
OK	Command has been accepted.

<b>Example:</b>	
IAA	Sets axis 1 to abort on a tracking error.

<b>AB</b>	<b>Abort</b>
-----------	--------------

The control of the motor may be aborted by sending **AB**. When aborted, the Error LED will illuminate the Error output will be activated, the servo loop is disabled and the motor shaft will be free to rotate. A user abort may be reset with the **RS** command. The position encoder is still read while aborted.

<b>Syntax</b>	<b>Units</b>	<b>Range</b>	<b>to</b>	<b>Initial State</b>
<ad>AB	N/A	N/A		N/A

<b>Condition Requirements</b>	<b>Notes:</b>
None.	The response to a <b>CO</b> command will be User Abort.

<b>Responses:</b>	
OK	Command has been accepted.

<b>Example:</b>	
1AB	Abort axis 1.

<b>AD</b>	<b>Toggle address prefix.</b>
-----------	-------------------------------

Toggle address prefix. Turns on or off (depending on the previous state) the axis address number attached to the beginning of a reply string. A colon : is added between the address and the reply.

<b>Syntax</b>	<b>Units</b>	<b>Range</b>	<b>to</b>	<b>Initial State</b>
<ad>AD	N/A	N/A		Prefix off

<b>Condition Requirements</b>	<b>Notes:</b>
None	Value retained on power-up.

<b>Response:</b>	
OK	Command has been accepted.

<b>Example:</b>	
1AD	Toggle address prefix on axis 1.
<b>Reply</b> 1:OK	
1AD	Toggle address prefix on axis 1.
<b>Reply</b> OK	

<b>AE</b>	<b>SET AUTO-EXECUTE SEQUENCE</b>
-----------	----------------------------------

Set sequence *n* to run on power-up (auto-execute) of the controller. This may be used in stand alone systems where there is no permanent host computer or terminal.

<b>Syntax</b>	<b>Units</b>	<b>Range</b>	<b>to</b>	<b>Initial State</b>
<ad>AEn	Seq. No.	0	7	Disabled

**Condition Requirements**  
None

**Notes:**  
Value retained on power-up.  
Cleared by Control-C and ESC.

**Responses:**

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.
! NO SEQUENCE	Sequence specified has not been defined yet.

**Example:**

1AE5      Sets auto execute of axis 1 to run sequence 5 on power-up.

<b>AL</b>	<b>ALLOW SOFT LIMITS</b>
-----------	--------------------------

Set the soft limit protection enable to ON. Further movement is bounded by the upper and lower soft limits. Soft limits may be turned OFF by the **IL** command.

<b>Syntax</b>	<b>Units</b>	<b>Range</b>	<b>to</b>	<b>Initial State</b>
<ad>AL	N/A	N/A		Enabled

**Condition Requirements**  
None.

**Notes:**  
Value retained on power-up.

**Response:**

OK	Command has been accepted.
----	----------------------------

**Example:**

1AL      Sets the soft limits ON for controller axis 1.



<b>AP</b>	<b>SET ACTUAL POSITION</b>
-----------	----------------------------

Set the actual position value to that given in the argument.

Syntax	Units	Range	to	Initial Value
<ad>APnnn	Steps	-2147483647	2147483647( $\pm 2^{32}$ )	N/A

Condition Requirements	Notes:
Idle or Constant velocity	Value zero on power-up.

Response:	
OK	Command has been accepted.

Examples:	
IAP5000	Set the axis 1 Actual Position to 5000.
or IAP0	Set the axis 1 Actual Position to zero.

<b>AR</b>	<b>ALLOW REMOTE (JOG) CONTROLS</b>
-----------	------------------------------------

Set the manual JOG control enable to ON. This enables movement by the JOG inputs. The JOG enable may be turned OFF by the **IR** command.

Syntax	Units	Range	to	Initial State
<ad>AR	N/A	N/A		Enabled.

Condition Requirements	Notes:
None	Value retained on power-up.

Response:	
OK	Command has been accepted.

Example:	
IAR	Sets the jog control enable ON for controller axis 1.

**CO****Display the Current Operation**

Output the current operation that the controller is executing, i.e. its status.

Syntax	Units	Range	to	Initial State
<ad>CO	N/A	N/A		N/A

**Condition Requirements**

None

**Notes:****Responses:**

Constant Velocity	Constant velocity move is executing.
Creep	Creep steps section of move is executing.
Delay	Time delay counter running.
Emergency Stop	Emergency Stop input active. No moves executing.
Execute Cam	Cam profile move executing.
Execute Profile	Timed profile move executing.
Idle	No moves executing.
Index	Index to datum executing.
Jog	Jog move executing.
Motor Stalled	<b>TH</b> (threshold) value exceeded due to position encoder failure or stalled motor. No moves executing.
Move	Move Absolute or Move Relative executing.
Settle	End of move settle time counter running.
Soft Stop	Decelerating to stop.
Synchronising	Waiting for position parity in absolute gearbox mode.
Tracking Abort	Controller is aborted due to exceeding the tracking window value <b>TR</b> . No moves executing.
User Abort	Controller is aborted due to the use of an <b>AB</b> (user abort) command. No moves executing.

**Example:**

IMR10000	Axis 1 move relative by 10,000 steps.
ICO	Query current operation for controller axis 1.

**Response:**

Move	Move Absolute or Move Relative executing.
------	---

<b>CP</b>	<b>SET COMMAND POSITION</b>
-----------	-----------------------------

Set the command position value to that given in the argument. The command position is the position generated by a move command.

Syntax	Units	Range	to	Initial Value
<ad>CPnnn	Steps	-2147483647	2147483647( $\pm 2^{32}$ )	N/A

Condition Requirements	Notes:
Idle or Constant velocity	Value zero on power-up.

Response:	
OK	Command has been accepted.

Examples:	
1CP5000	Set the axis 1 Command Position to 5000.
or 1CP0	Set the axis 1 Command Position to zero.

<b>CR</b>	<b>SET CREEP DISTANCE</b>
-----------	---------------------------

Set number of creep steps at the end of a move. The motor will decelerate and execute this number of steps at the creep speed.

Syntax	Units	Range	to	Initial Value
<ad>CRnnn	Steps	0	2147483647( $2^{32}$ )	0

Condition Requirements	Notes:
Idle or Constant velocity	Value retained on power-up.

Response:	
OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Examples:	
1CR50	Set the creep distance to 50 steps on axis 1.

<b>CV</b>	<b>Constant Velocity Move</b>
-----------	-------------------------------

A Constant velocity move ramps up at **SA** acceleration rate, then moves the motor at the set **SV** speed. The **SV** command is used to change the speed whilst motion is in progress. The **SA** and **SD** rates define the rate at which the change of speed will be made. Constant velocity mode is exited by an **ST** command, **ESC** or Control **C**.

Syntax	Units	Range	to	Initial Value
<ad>CVn	Direction	-ve	+ve	N/A
<b>Condition Requirements</b>		<b>Notes:</b>		
Idle.		During a constant velocity move, <b>SA</b> , <b>SD</b> and <b>SV</b> commands are acted upon immediately.		
<b>Responses:</b>				
OK		Command has been accepted.		
! EMERGENCY STOP		The Emergency Stop has been activated.		
! TRACKING ABORT		Controller has aborted due to a Tracking error.		
! USER ABORT		Controller is aborted due to a user command.		
! MOTOR STALLED		Controller is aborted due to stalled motor or encoder loss.		

<b>Examples:</b>		
1CV	Start constant velocity move in positive direction on axis 1.	
1CV-1	Start constant velocity move in negative direction on axis 1.	

<b>DA</b>	<b>Difference Actual Position</b>
-----------	-----------------------------------

Add value to actual (position encoder's) position.

Syntax	Units	Range	to	Initial Value
<ad>DAnnn	Steps	-2147483648	2147483647 ( $\pm 2^{32}$ )	N/A
<b>Condition Requirements</b>		<b>Notes:</b>		
None.				
<b>Response:</b>				
OK		Command has been accepted.		
<b>Examples:</b>				
1OA	Get the axis 1 Actual Position.			
Response:	15000			
1DA5000	Difference axis 1 actual position by 5000.			
1OA	Get the axis 1 Actual Position.			
Response:	20000			

DB	Set Deadband
----	--------------

Set *Deadband* in number of steps either side of command position. The output from the controller will not increase or decrease for position errors less than the value of the *deadband*.

Syntax	Units	Range	to	Initial Value
<ad>DBnnn	Steps	0	4000	0

Condition Requirements	Notes:
None.	Value retained on power-up. The value of <i>deadband</i> is set to zero by the <b>TUNE</b> command.

Responses	
OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

**Example:**  
1DB5                      Set deadband on axis 1 to 5 steps.

DC	Difference Command Position
----	-----------------------------

Add value to Command Position.

Syntax	Units	Range	to	Initial Value
<ad>DCnnn	Steps	-2147483648	2147483647 ( $\pm 2^{32}$ )	N/A

Condition Requirements	Notes:
None.	

Response:	
OK	Command has been accepted.

**Examples:**  
1OC                      Get the axis 1 Command Position.  
Response:              15000  
  
1DC5000                Difference axis 1 Command position by 5000.  
1OC                      Get the axis 1 Command Position.  
Response:              20000



DE	DELAY
----	-------

This command will start a delay timer for the time given in the argument. After completion of this time, the controller will return to the idle mode.

If the next command should not execute until the end of this delay time, and is not a *wait for idle* command, then the **DE** command must be followed with a Wait for End (**WE**) command. This will make the controller wait until it returns to the *idle* state before executing the next command.

Syntax	Units	Range	to	Initial Value
<ad>DEnnn	milliseconds	0	60000	N/A

#### Condition Requirements

Idle and not Constant velocity or Cam.

#### Notes:

#### Responses:

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.
! TRACKING ABORT	Controller has aborted due to a Tracking error.
! USER ABORT	Controller is aborted due to a user command.
! EMERGENCY STOP	The Emergency Stop has been activated.
! MOTOR STALLED	Controller is aborted due to stalled motor or encoder loss.
! CONTEXT	Not available in CV, gearbox or Cam modes.

#### Examples:

1MR400	Move 400 steps positive.
1DE2000	Delay for 2 seconds then....
1MR-400	Move 400 steps negative.
1WP2220	Turn LED on (write port 1).
1DE1000	Delay for 1 second.
1WE	Wait for End of Delay
1WP2221	Turn LED off (write port 1).

<b>DP</b>	<b>DEFINE PROFILE</b>
-----------	-----------------------

This command will start a Profile definition. The only command that is used during a Profile definition is **MR** any other commands except for **EP** will cause a ! SYNTAX error.

The commands that follow this **DP** command will not be executed, but will be stored in the on board non-volatile memory until the End Profile definition (**EP**) command is received. If a Control-C or ESCAPE command is received or the controller runs out of memory, the Profile definition will cease, the Profile will not be stored and the controller will return to the idle state.

Syntax	Units	Range	to	Initial Values
<ad>DP	N/A	N/A		N/A

**Condition Requirements**  
Idle.

**Notes:**  
If a Profile is defined, that fact is shown on the **QA** page.

**Responses:**

OK	Command has been accepted.
! RECURSIVE DP	DP attempted when already defining a Profile.
! MEMORY OFLO	The available memory has overflowed.
! DP SYNTAX	Command is not <b>MR</b> or <b>EP</b> .

**Example:**

IDP	Start Profile definition.
1MR200	First Profile move.
1MR500	Next Profile move.
1MR-500	"
1MR-200	"
1MR50	"
1EP	End of Profile definition.

<b>DS</b>	<b>DEFINE SEQUENCE</b>
-----------	------------------------

This command will start a sequence definition. There are eight sequences that may be defined and the argument selects which sequence is to be defined (0 to 7).

All valid commands that follow this **DS** command will not be executed, but will be stored in the on board non-volatile memory until the End Sequence definition (**ES**) command is received. If a Control-C or ESCAPE command is received or the controller runs out of memory, the sequence definition will cease, the sequence will not be stored and the controller will return to the idle state.

Syntax	Units	Range	to	Initial Values
<ad>DSn	Seq. No.	0	7	N/A

Condition Requirements	Notes:
Idle	Auto-execute flag is cleared. The sequences defined are shown on the <b>QA</b> page.

Responses	
OK	Command has been accepted.
! OUT OF RANGE	Argument (sequence number) is out of valid range.
! RECURSIVE DS	DS attempted when already defining a sequence.
! MEMORY OFLO	The available memory has overflowed.

Example:	
IDS4	Start definition of sequence 4.
ISV2000	Set slew speed.
IMA8000	First move (absolute).
IMR5000	Next move (relative).
IMR3000	Next move (relative).
ISV20000	Set new slew speed.
IMA0	Next move (return to start position).
IXS4	Execute sequence 4 (loop to start of this sequence).
IES	End of sequence definition.

<b>EP</b>	<b>END PROFILE DEFINITION</b>
-----------	-------------------------------

This command will end a Profile definition. The Profile definition must have been started by the Define Profile (**DP**) command.

Syntax	Units	Range	to	Initial Value
<ad>EP	N/A	N/A		N/A

Condition Requirements	Notes:
Define Profile	

Responses:	
OK	Command has been accepted.
! EP WITHOUT DP	EP attempted when NOT already defining a Profile.

Example:	
IDP	Start Profile definition.
IMR200	First Profile move.
IMR500	Next Profile move.
IEP	End of Profile definition.

<b>ES</b>	<b>END SEQUENCE DEFINITION</b>
-----------	--------------------------------

This command will end a sequence definition. The sequence definition must have been started by the Define Sequence (DS) command. No argument is necessary as the sequence number is specified with the Define Sequence (DS) command.

Syntax	Units	Range	to	Initial Value
<ad>ES	N/A	N/A		N/A

Condition Requirements	Notes:
Define Sequence	

Responses	
OK	Command has been accepted.
! ES WITHOUT DS	ES attempted when NOT already defining a sequence.

Example:	
1DS2	Start definition of sequence 2.
1MR400	First move (relative).
1MR-400	Next move (relative).
1XS5	Execute sequence 5 (transfer control to start of this sequence 5).
1ES	End of sequence definition.

Help pages. The commands **HE1** and **HE2** return pages showing Digiloop commands. These help pages give a concise list of the commands available and their function.

Syntax	Units	Range	to	Initial Value
<ad>HEn	page	0	2	N/A

Condition Requirements	Notes:
None.	The command HE0 is the same as HE1. Values of HE greater than 1 show HE2.

**Responses:**

<b>HE1</b>			
AA	allow abort	AB	abort move
AD	toggle addressing	AE<seq no.>	autoexecute sequence
AL	allow limits	AP<position>	set actual position
AR	allow jog	CO	current operation
CP<position>	set command position	CR<distance>	set creep steps
CV<direction>	constant velocity	DA<difference>	actual position
DB<distance>	set deadband	DC<difference>	command position
DE<time in ms>	delay	DP .. EP	define profile
DS<seq no.>.. ES	define sequence	HE1, HE2	help
ID	identify	IN	initialise
IA	ignore abort	IL	ignore limits
IR	inhibit jog	IX<direction>	index
KF<value>	set feedforward co.	KP<value>	set proportional co.
KS<value>	set sum co.	KV<value>	set velocity co.
LL<position>	set lower soft limit		
<b>HE2</b>			
MA<position>	move absolute	MR<distance>	move relative
OA	output actual pos	OC	output command pos
OD	output difference	OS	output status string
QA	query all	QK	query coefficients
QS	query speeds	RP	read port
RS	reset from abort	RSES	reset from emg. stop
RSST	reset from stalled	SA<acceleration>	set acceleration
SD<deceleration>	set deceleration	SE<time in ms>	set settling time
SC<speed>	set creep speed	SN<bit pattern>	skip on condition
ST	stop	SV<speed>	set velocity
TH<value>	set stall threshold	TR<distance>	set tracking window
TUNE	tune coefficients	UL<position>	set upper soft limit
UP, US	undefine cam etc.	VP, VS	verify cam etc.
WA<bit pattern>	wait for condition	WE	wait for end of move
WI<distance>	set window	WP<bit pattern>	write to port
XP<time in ms>	execute profile	XS<seq no.>	execute sequence

**Example:**  
1HE1            Show the first help page of the controller of axis 1.



<b>IA</b>	<b>IGNORE ABORT</b>
-----------	---------------------

Ignore Tracking abort. If the error between the command position and actual position exceeds the **TR** Tracking window value, the controller does not abort but continues to control the motor in the normal way. The Error signal and LED are still activated for the duration of the error. If the size of the tracking window is altered when in this mode, the error message ! TRACKING DISABLED will be returned. The setting of the Tracking Abort, either Enabled or Disabled is shown in the **QA** page.

<b>Syntax</b>	<b>Units</b>	<b>Range</b>	<b>to</b>	<b>Initial Value</b>
<ad>IA	N/A	N/A		Enabled

<b>Condition Requirements</b>	<b>Notes:</b>
None.	Value retained on power-up.

<b>Response:</b>	
OK	Command has been accepted.

<b>Example:</b>	
11A	Tracking errors on axis 1 will not cause an abort.

<b>ID</b>	<b>IDENTIFY</b>
-----------	-----------------

This command is used to give the type of controller and its internal software revision.

<b>Syntax</b>	<b>Units</b>	<b>Range</b>	<b>to</b>	<b>Initial Value</b>
<ad>ID	N/A	N/A		N/A

<b>Condition Requirements</b>	<b>Notes:</b>
None	

<b>Response:</b>	
	Mclennan Servo Supplies Ltd. PM300 V6.12

<b>Example:</b>	
1AD	Toggle address prefix on or off.
11D	Identify controller of axis 1. If this was a PM300 and the address prefix was previously off, it would respond: 1:Mclennan Servo Supplies Ltd. PM300 V6.12

<b>IL</b>	<b>IGNORE SOFT LIMITS</b>
-----------	---------------------------

Set the soft limit protection enable to OFF. Further movement is NOT bounded by the upper and lower soft limits. Soft limits may be turned ON by the **AL** (allow limits) command. Hard limits will still be active and cannot be disabled.

<b>Syntax</b>	<b>Units</b>	<b>Range</b>	<b>to</b>	<b>Initial Value</b>
<ad>IL	N/A	N/A		Enabled.

<b>Condition Requirements</b>	<b>Notes:</b>
None.	Value retained on power-up.

<b>Response:</b>	
OK	Command has been accepted.

<b>Example:</b>	
1IL	Sets the soft limits OFF for controller axis 1.

<b>IN</b>	<b>INITIALISE</b>
-----------	-------------------

This command will set all the programmable parameters back to their initial values, clear sequences and profiles. This is used to re-initialise all the non-volatile memory values to 'safe' values - e.g. if the controller was to be used in a new application.

<b>Syntax</b>	<b>Units</b>	<b>Range</b>	<b>to</b>	<b>Initial Value</b>
<ad>IN	N/A	N/A		N/A

<b>Condition Requirements</b>	<b>Notes:</b>
None.	

<b>Response:</b>	
OK	Command has been accepted.

<b>Example:</b>	
1IN	Set all parameters on axis 1 back to their initial values.

<b>IR</b>	<b>INHIBIT REMOTE (JOGS) CONTROLS</b>
-----------	---------------------------------------

Disable movement by the JOG inputs. The JOG inputs may be re-enable by the **AR** (allow remote) command.

<b>Syntax</b>	<b>Units</b>	<b>Range</b>	<b>to</b>	<b>Initial Value</b>
<ad>IR	N/A	N/A		Enabled.

<b>Condition Requirements</b>	<b>Notes:</b>
None	Value retained on power-up.

<b>Response:</b>	
OK	Command has been accepted.

<b>Example:</b>	
1IR	Disable the Jog control inputs for controller on axis 1.

<b>IX</b>	<b>FIND INDEX MARKER (DATUM)</b>
-----------	----------------------------------

This command is used to find a datum point of a mechanism.

Index to datum ramps the motor up at **SA** rate, then moves at the set **SV** rate until a *Slow down* (datum approach) signal is received on read port 1. It will then ramp down at **SA** rate to **SC** creep speed until receipt of a *Stop* (datum stop) signal on Read Port 2. User may then define this position as required.

**Important.** The respective **SV** and **SA** rates and index sensor positions should be set such that the creep speed is reached before the final stop signal is received. For accurate location of the *datum* position the creep speed **SC** should be set at below 1000 steps per second. After an index operation the positions are not reset.

If no Datum Approach switch is used then the search is performed only at the creep speed.

The **IX-1** command will perform the same in the negative direction.

Soft limits are **not** used during an Index mark search.

Syntax	Units	Range	to	Initial Value
<ad>IXnnn	N/A	-ve	+ve	N/A

Condition Requirements	Notes:
Idle.	

Responses:	
OK	Command has been accepted.
! HARD LIMIT	Move attempted when already on hard limit.
! TRACKING ABORT	Controller has aborted due to a Tracking error.
! USER ABORT	Controller is aborted due to a user command.
! EMERGENCY STOP	The Emergency Stop has been activated.
! MOTOR STALLED	Controller is aborted due to stalled motor or encoder loss.

Examples:	
IIX	Search for datum point of axis 1 in positive direction.
IIX-1	Search for datum point of axis 1 in negative direction.

<b>KF</b>	<b>Set Feedforward Coefficient</b>
-----------	------------------------------------

Set velocity feedforward servo coefficient. This compensates for the position offset caused by the velocity lag introduced by **KV**. For positioning moves **KF** is normally set at zero, but for Profiles and Cam moves where the actual position should not lag behind the command position, **KF** should be set equal to **KV**.

It is not usually necessary for complete compensation of the velocity lag as this adversely effects the settling time of the system.

Syntax	Units	Range	to	Initial Value
<ad>KFnnn	Number	0	32767	0

Condition Requirements	Notes:
None.	Value retained on power-up.

Responses	
OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Examples:	
1KF500	Set velocity feedforward on axis 1 to 500.

<b>KP</b>	<b>Set Proportional Gain Coefficient</b>
-----------	--

Set proportional gain servo coefficient. The stiffness and accuracy of the servo loop are controlled by the magnitude of the proportional gain.

Syntax	Units	Range	to	Initial Value
<ad>KPnnn	Number	0	32767	10

Condition Requirements	Notes:
None.	Value retained on power-up.

Responses	
OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

**Examples:**  
1KP100      Set the proportional gain on axis 1 to 100.

<b>KS</b>	<b>Set Sum Gain Coefficient</b>
-----------	---------------------------------

The Sum servo coefficient is the sum of the integral and proportional components of the servo control loop. The accuracy of the servo loop depends on having a non-zero value of **KS** at the expense of transient response.

Syntax	Units	Range	to	Initial Value
<ad>KSnnn	Number	0	32767	0

Condition Requirements	Notes:
None.	Value retained on power-up.

Responses	
OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

**Examples:**  
1KS50      Set the Sum gain on axis 1 to 50.

<b>KV</b>	<b>Set Velocity Feedback Coefficient</b>
-----------	--

The value of this coefficient defines the magnitude of the velocity feedback signal derived from the position encoder. This coefficient influences the transient response by producing a damping effect. It effects the system by reducing overshoot and enhancing stability, but too high a value can create a *buzzy* system, and ultimately an unstable system.

<b>Syntax</b>	<b>Units</b>	<b>Range</b>	<b>to</b>	<b>Initial Value</b>
<ad>KVnnn	Number	0	32767	0

<b>Condition Requirements</b>	<b>Notes:</b>
None.	Value retained on power-up.

<b>Responses</b>	
OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

**Examples:**  
1KV500            Set the Velocity feedback on axis 1 to 500.

<b>LL</b>	<b>SET LOWER SOFT LIMIT POSITION</b>
-----------	--------------------------------------

This command will set the Lower Soft Limit Position to the value given in the argument. Subsequent moves by the Move Absolute (**MA**) or Move Relative (**MR**) and manual Jog moves will not be allowed below this Lower Limit if the Soft Limits are enabled.

<b>Syntax</b>	<b>Units</b>	<b>Range</b>	<b>to</b>	<b>Initial Value</b>
<ad>LLnnn	Steps	-2147483648	2147483647 ( $\pm 2^{32}$ )	-2147483648 ( $-2^{32}$ )

<b>Condition Requirements</b>	<b>Notes:</b>
Idle or Constant velocity	Value retained on power-up.

<b>Responses</b>	
OK	Command has been accepted.
! LIMITS DISABLED	A warning that the soft limits are currently disabled.
! LIMIT CONFLICT	Attempting to set lower limit above or equal to upper limit

**Example:**  
ILL-4000            Set the axis 1 Lower Soft Limit Position to -4000.



<b>MA</b>	<b>MOVE TO ABSOLUTE POSITION</b>
-----------	----------------------------------

This command will move the motor to the position given in the argument. This position is relative to the Command Position of zero.

Syntax	Units	Range	to	Initial Value
<ad>MAnnn	Steps	-2147483647	2147483647 ( $\pm 2^{32}$ )	N/A

Condition Requirements	Notes:
Idle	

Responses	
OK	Command has been accepted.
! EMERGENCY STOP	The Emergency Stop has been activated.
! TRACKING ABORT	Controller has aborted due to a Tracking error.
! USER ABORT	Controller is aborted due to a user command.
! MOTOR STALLED	Controller is aborted due to stalled motor or encoder loss.
! HARD LIMIT	Move attempted when already on a hard limit.
! SOFT LIMIT	Move attempted beyond a soft limit.

**Example:**  
If axis 1 has a current Command Position of 5000 then the command:  
    1MA4000      Will move 1000 steps in the negative direction to arrive at a Command position of 4000.

<b>MR</b>	<b>MOVE TO RELATIVE POSITION</b>
-----------	----------------------------------

This command will move the motor to the position given in the argument relative to the current Command Position.

Syntax	Units	Range	to	Initial Value
<ad>MRnnn	Steps	-2147483647	2147483647 ( $\pm 2^{32}$ )	N/A

Condition Requirements	Notes:
Idle	

Responses	
OK	Command has been accepted.
! EMERGENCY STOP	The Emergency Stop has been activated.
! TRACKING ABORT	Controller has aborted due to a Tracking error.
! USER ABORT	Controller is aborted due to a user command.
! MOTOR STALLED	Controller is aborted due to stalled motor or encoder loss.
! HARD LIMIT	Move attempted when already on a hard limit.
! SOFT LIMIT	Move attempted beyond a soft limit.

**Example:**  
If axis 1 has a current Command Position of 5000 then the command:  
    1MR4000      Will move 4000 steps in the positive direction to arrive at a Command position of 9000.

<b>OA</b>	<b>OUTPUT ACTUAL POSITION</b>
-----------	-------------------------------

This command will give the current encoder read Actual Position. This position is derived from the incoming position encoder pulses.

Syntax	Units	Range	to	Initial Value
<ad>OA	N/A	N/A		N/A

<b>Condition Requirements</b>	<b>Notes:</b>
None	

**Response:**  
The response is a string of numeric characters.

**Example:**  
If the controller of axis 1 currently has an Actual Position of 20501 then the command:  
1OA                      will respond:    AP=20501

<b>OC</b>	<b>OUTPUT COMMAND POSITION</b>
-----------	--------------------------------

This command will give the current Command Position.

Syntax	Units	Range	to	Initial Value
<ad>OC	N/A	N/A		N/A

<b>Condition Requirements</b>	<b>Notes:</b>
None.	

**Response:**  
The response is a string of numeric characters.

**Example:**  
If the controller of axis 1 currently has a Command Position of 45280 then the command:  
1OC                      will respond:    CP=45280

<b>OD</b>	<b>OUTPUT DIFFERENCE BETWEEN COMMAND AND ACTUAL POSITIONS</b>
-----------	---

This command will give the difference between the current Command Position and the current encoder read Actual Position. Numerically it is the Command Position (**CP**) - Actual Position (**AP**).

Syntax	Units	Range	to	Initial Value
<ad>OD	N/A	N/A		N/A

**Condition Requirements**                      **Notes:**  
None.

**Response:**  
The response is a string of numeric characters.

**Example:**  
If the controller of axis 1 currently has a Current position of 1000 and an Actual Position of 1050 then the command:  
IOD                      will respond:     DP=- 50

<b>OS</b>	<b>OUTPUT STATUS STRING</b>
-----------	-----------------------------

This command will return an eight bit string that indicates the status of the controller in a format that is more easily interpreted by a host computer.

Syntax	Units	Range	to	Initial Value
<ad>OS	N/A	N/A		N/A

**Condition Requirements**                      **Notes:**  
None.

**Responses:**  
The response is a string of four numeric characters. The characters are either '0' for not active or '1' for active. The 1: part would only appear if the **AD** address toggle had been set to prefix replies with the axis address of the replying controller.

```

1:00000000
| | | | | | | | Emergency Stop: 1 = active.
| | | | | | | | Motor Stalled: 1 = active.
| | | | | | | | Tracking Abort: 1 = active.
| | | | | | | | User Abort: 1 = active.
| | | | | | | | Controller Idle, i.e., awaiting next command: 1 = idle.
| | | | | | | | Not Error: 1 = not stopped, not stalled nor aborted.
| | | | | | | | +ve Hard Limit: 1 = activated.
| | | | | | | | -ve Hard Limit: 1 = activated.
```

**Example:**  
If the PM300 on axis 1 currently is idle, not stopped, stalled, aborted nor on either hard limit, then the command:  
IOS                      will respond:     00110000

<b>QA</b>	<b>QUERY ALL PARAMETERS</b>
-----------	-----------------------------

Query All. Returns all of the current settings and modes of the controller along with the current positions in a single page format.

Syntax	Units	Range	to	Initial Value
<ad>QA	N/A	N/A		N/A

<b>Condition Requirements</b>	<b>Notes:</b>
None.	

**Response:**  
The response is alpha-numeric strings of characters. Each line gives the parameter name and its value. See example for the format.

**Example:**  
1QA Will generate a response of the form:

McLennan Servo Supplies Ltd PM300 V6.12	
Address: 1	Address Echo: Enabled
Status: Idle	
KP=2132	KS=2304 KV=370 KF=370
Slew Speed = 200000	
Acceleration = 10000	Deceleration = 10000
Creep Speed = 100	Creep Steps = 0
Jog Speed = 100	Fast Jog = 500
Settling = 10	Deadband = 0
Window = 4	Threshold = 200
Tracking Abort: Enabled	Tracking = 4000
Soft Limits: Enabled	Jog: Enabled
Lower Limit = -2147483648	Upper Limit = 2147483647
Lower hard limit: Off	Upper hard limit: Off
Gbox Num = 1	Gbox Den = 1
Command Pos = 98789	Actual Pos = 98789
Pos Error = 0	Input Pos = -189
Autoexec: Sequence #6	
Sequences: 0,1,2,3,4,5,6	No Profile
Cam Defined	Cam modulo = 8000
Memory Usage 95%	
Read Port: 1111	Last Write: 1111

<b>QK</b>	<b>QUERY K COEFFICIENTS</b>
-----------	-----------------------------

Query servo loop coefficients. Returns the current settings of the KP, KS, KV, and KF coefficients.

Syntax	Units	Range	to	Initial Value
<ad>QK	N/A	N/A		N/A

Condition Requirements	Notes:
None	

**Response:**  
The response is an alpha-numeric string of characters showing the parameter name and its value. See example for the format.

<b>Example:</b>	
1IN	Set to initial values.
1KP2909	Set proportional gain to 2909.
1KV357	Set velocity feedback to 357.
1KS3258	Set Sum coefficient to 3258.
1QK	Will generate a response of the form: KP=2909,KS=3258,KV=357,KF=0

<b>QS</b>	<b>QUERY SPEEDS</b>
-----------	---------------------

Query the current settings for the speeds and accelerations. Returns the current settings of SV, SC, SA and SD.

Syntax	Units	Range	to	Initial Value
<ad>QS	N/A	N/A		N/A

Condition Requirements	Notes:
None	

**Response:**  
The response is an alpha-numeric string of characters showing the parameter name and its value. See example for the format.

<b>Example:</b>	
1SC1000	Set creep speed to 1000 steps/sec.
1SV16200	Set slew speed to 16200 steps/sec.
1SA100000	Set deceleration to 100,000 steps/sec <sup>2</sup> .
1SD100000	Set deceleration to 100,000 steps/sec <sup>2</sup> .
1QK	Will generate a response of the form: SV=16200,SC=1000,SA=100000,SD=100000



<b>RP</b>	<b>READ INPUT PORT</b>
-----------	------------------------

This command will examine the read port inputs and return their current state as a four digit numeric string of either 0 or 1 characters. The string starts with read port 4. A 1 indicates that the input is low (0V or open-circuit) and a 0 indicates that the input is high (+24V).

Syntax	Units	Range	Initial Value
<ad>RP	N/A	N/A	N/A

#### Condition Requirements

None

#### Notes:

If an **RP** command is executed with the read ports open circuit, a reply of 1111 will be returned

#### Responses

A four digit numeric string.

#### Example:

If the following states are present on the inputs:

PORT :	4	3	2	1
STATE :	Low	Low	Low	High

then the command 1RP will reply: 1110

<b>RS</b>	<b>RESET FROM ABORT</b>
-----------	-------------------------

This command will reset the *tracking abort* or *user abort* conditions and re-enable the servo control loop. It will also set the Command position to be equal to the Actual position.

Syntax	Units	Range	Initial Value
<ad>RS	N/A	N/A	N/A

#### Condition Requirements

None.

#### Notes:

#### Responses

OK	Command has been accepted.
! EMERGENCY STOP	The Emergency Stop has been activated.
! MOTOR STALLED	Controller is aborted due to stalled motor or encoder loss.
! NOT ABORTED.	The unit has not aborted.

#### Example:

1RS                      Reset abort on axis 1 controller.

<b>RSES</b>	<b>RESET FROM EMERGENCY STOP</b>
-------------	----------------------------------

This command will, if the emergency stop input is not active, resets the *Emergency Stop* condition and re-enables the servo control loop. It will also set the Command position to be equal to the Actual position.

Syntax	Units	Range	Initial Value
<ad>RSES	N/A	N/A	N/A

<b>Condition Requirements</b>	<b>Notes:</b>
None.	

<b>Response</b>	
OK	Command has been accepted.
! NOT STOPPED	Stop Input has not been activated.

**Example:**  
1RSES            Reset stopped condition on axis 1 controller.

<b>RSST</b>	<b>RESET FROM MOTOR STALLED</b>
-------------	---------------------------------

This command will reset the *Motor Stalled* condition and re-enable the servo control loop. It will also set the Command position to be equal to the Actual position.

Syntax	Units	Range	Initial Value
<ad>RSST	N/A	N/A	N/A

<b>Condition Requirements</b>	<b>Notes:</b>
None.	

<b>Response</b>	
OK	Command has been accepted.
! NOT STALLED	Motor Stalled threshold has not been exceeded.

**Example:**  
1RSST            Reset motor stalled condition on axis 1 controller.

<b>SA</b>	<b>SET ACCELERATION</b>
-----------	-------------------------

Set the acceleration rate for changes of velocity for all following moves. This may also be used during a constant velocity move.

<b>Syntax</b>	<b>Units</b>	<b>Range</b>	<b>to</b>	<b>Initial Value</b>
<ad>SA $\overline{nnn}$	Steps/sec <sup>2</sup>	1	20000000	10000

**Condition Requirements**  
Idle or Constant velocity

**Notes:**  
Value retained on power-up.

**Responses:**

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

**Example:**

1SA10000      Sets acceleration of axis 1 controller to 10000 Steps/sec<sup>2</sup>.

<b>SC</b>	<b>SET CREEP SPEED</b>
-----------	------------------------

Set the creep speed for all following moves. This is the speed that at which moves with a non-zero creep distance will stop.

It is also the speed that slow datum search will be moved at (**IX** command).

<b>Syntax</b>	<b>Units</b>	<b>Range</b>	<b>to</b>	<b>Initial Value</b>
<ad>SC $\overline{nnn}$	Steps/sec	1	400000	100

**Condition Requirements**  
Idle or Constant velocity

**Notes:**  
Value retained on power-up.

**Responses**

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

**Example:**

1SC700      Sets creep speed of axis 1 controller to 700 Steps/sec.

<b>SD</b>	<b>SET DECELERATION</b>
-----------	-------------------------

Set the deceleration rate for changes of velocity for all following moves. This may also be used during a constant velocity move.

Syntax	Units	Range	to	Initial Value
<ad>SDnnn	Steps/sec <sup>2</sup>	1	20000000	10000

**Condition Requirements**  
Idle or Constant velocity

**Notes:**  
Value retained on power-up.

<b>Responses:</b>	
OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

**Example:**  
1SD100000      Sets acceleration of axis 1 controller to 100000 Steps/sec<sup>2</sup>.

<b>SE</b>	<b>SET SETTLING TIME</b>
-----------	--------------------------

Set the settling time for all following moves. This time elapses at the end of each move to allow the motor to settle. The end of a move is defined by the **OD** (position difference) value being less than the **WI** (end of move window) value for the **SE** (settling) time.

Syntax	Units	Range	to	Initial Value
<ad>SEnnn	milliseconds	0	20000	10

**Condition Requirements**  
Idle or Constant velocity

**Notes:**  
Value retained on power-up. While the settling time is elapsing the **CO** command will give a reply of **Settle**.

<b>Responses</b>	
OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

**Example:**  
1SE1000      Sets settling time of axis 1 controller to 1 second.

SN	SKIP NEXT
----	-----------

Skip next command if true. The controller will *skip over* (ignore) the following command if the read ports correspond to the bit pattern specified. This command will examine the read port inputs and compare them with the specified bit pattern argument. If the inputs are equal to the specified bit pattern, then the controller will skip over, i.e. not execute the next command. If no commands are in the command buffer or in a sequence the *next* command will be the next received command. If the *skip* condition is not met, then the next command will be executed as normal. If the next command is skipped, the controller will give the response SKIPPED instead of OK or any other response for that command.

The bit pattern is specified as a four digit binary number of either 0, 1 or 2 characters starting with read port 4, through to 1. A 0 defines that the input must be high (+24V), a 1 defines that the input must be low (0V) and a 2 defines that the input is not relevant or *don't care*. If less that four digits are specified in the argument, then the preceding ones are assumed as high (0).

This command may be used to introduce a conditional response to some machine functions, and may be used to create *smart* sequences.

Syntax	Units	Range	Initial Value
<ad>SNbbbb	Bit pattern	4 digits of 0, 1 or 2	N/A

Condition Requirements	Notes:
None.	

Responses	
OK	Command has been accepted.
! SN SYNTAX	Invalid argument i.e. bit specified was not 0, 1 or 2 OR the Number of bits was greater than 4.

Example:

1DS3	Define the start of the sequence 3.
1WA2221	Wait here until read port 1 goes low.
1SN2212	Skip next command if read port 2 is low, state of ports 1, 3 & 4 not important.
1IX	Search for datum (only executed if port 2 was high, above).
1MR1000	Move motor 1000 steps.
1XS	Loop back to start of sequence.
1ES	End definition of sequence.

The sequence shown is a repeated incremental indexing motion, but allows an operator to search for datum only when required, usually just after power-up. If the **AE** (Auto execute) flag is set, the system could be operated without the presence of a host computer.



<b>ST</b>	<b>STOP</b>
-----------	-------------

This command will stop any current move, decelerate the motor speed down at the **SD** rate, then stop and return to *idle* mode.

This command is buffered and is only responded to when it reached in the command queue. Care must therefore be taken that there are no commands that hold up the queue between the move command and the **ST** command.

<b>Syntax</b>	<b>Units</b>	<b>Range</b>	<b>to</b>	<b>Initial Value</b>
<ad>ST	N/A	N/A		N/A

<b>Condition Requirements</b>	<b>Notes:</b>
None	Will exit constant velocity mode or gearbox mode.

<b>Responses</b>	
OK	Command has been accepted.
! EMERGENCY STOP	The Emergency Stop has been activated.
! MOTOR STALLED	Controller is aborted due to stalled motor or encoder loss.

**Example:**

<b>1CV</b>	Will start axis 1 moving in constant velocity mode.
<b>1ST</b>	This will then stop the current move of axis 1.

<b>SV</b>	<b>SET VELOCITY</b>
-----------	---------------------

Set the Slew (maximum) velocity for all following moves. This may also be used during a constant velocity move.

<b>Syntax</b>	<b>Units</b>	<b>Range</b>	<b>to</b>	<b>Initial Value</b>
<ad>SVnnn	Steps/sec	1	400000	100

<b>Condition Requirements</b>	<b>Notes:</b>
Idle or Constant velocity	Value retained on power-up.

<b>Responses</b>	
OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

**Example:**

<b>1SV5000</b>	Sets slew speed of axis 1 controller to 5000 Steps/sec.
----------------	---

<b>TH</b>	<b>SET THRESHOLD</b>
-----------	----------------------

This command will set the motor stalled threshold. Failure of an encoder is indistinguishable from a stalled motor, and messages from the PM300 refer to *motor stalled* rather than encoder failure.

A stalled motor (or encoder failure) is detected by looking for changes in the position encoder signals (or equivalently the changes in observed motor position). If the motor does not move, and the voltage output value from the PM300 exceeds the value set by the **TH** command for a time of 256ms, then the PM300 will set its output to zero and set a Motor Stalled condition..

The servo system will have coulomb friction and the voltage required to overcome this friction, varies from system to system, so the value of **TH** must be large enough not to nuisance trigger but small enough to detect any failure.

If a *stalled motor* condition occurs, the error signal and front panel LED are both activated, and movement is stopped. Subsequent moves will not function but will return the response ! MOTOR STALLED until reset by either a Reset Stall (**RSST**) command or by powering off.

The response to a **CO** command is **Motor Stalled**.

Syntax	Units	Range	to	Initial Value
<ad>THnnn	Steps	0	2047	200

Condition Requirements	Notes:
None.	Value retained on power-up.

Responses:	
OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

**Example:**  
I' TH400            Set the Threshold before *motor stalled* condition for axis I to 400.

<b>TR</b>	<b>SET TRACKING WINDOW</b>
-----------	----------------------------

This command will set the tracking window. The *Tracking window* is the allowable difference between the *Command Position* and the *Actual Position*. When the motor is stationary this is the allowable static error. During a move, a changing *command position* is generated. The *Tracking Window* operates on the difference between the *actual position* and this moving *command position*. The servo system will have a *following error*, so the value of **TR** must be large enough not to nuisance trigger but small enough to detect any failure.

If the *tracking window* is exceeded the Error output signal and front panel LED are activated and (if abort is enabled) the controller *aborts*.

The abort function may be inhibited by using the **IA** (ignore abort) command, or enabled using the **AA** (allow abort) command.

Subsequent moves will not function but will return the response ! TRACKING ABORT until reset by either a Reset (**RS**) command or by powering off.

Syntax	Units	Range	to	Initial Value
<ad>TRnnn	Steps	0	2147483647 (2 <sup>32</sup> )	4000

Condition Requirements	Notes:
None.	Value retained on power-up.

Responses:	
OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.
! TRACKING DISABLED	Warning that tracking abort is inhibited. Value will be accepted.

**Example:**  
1TR400            Set the Tracking Window for axis 1 to 400 steps.

<b>TUNE</b>	<b>TUNE COEFFICIENTS</b>
-------------	--------------------------

An approximate set of servo coefficients can usually be derived by invoking the **TUNE** command. The controller will *exercise* the motor over a small displacement for a few seconds and obtain a set of values for the *K* coefficients that should be stable and provide a reasonable disturbance rejection.

The tuning algorithm may fail if there is excessive backlash, if the low frequency loop gain is either very small or very large or the feedback encoder phasing is wrong. Further optimisation of system response will be required in almost all cases to achieve the desired performance.

The **TUNE** command only affects **KP, KV, KS** and **KV** therefore its use in a double encoder system is inappropriate and may produce a **! TUNE FAILURE** error.

Syntax	Units	Range	to	Initial Value
<ad>TUNE	N/A	N/A	N/A	N/A

Condition Requirements	Notes:
Idle.	Value retained on power-up. Deadband value set to zero.

Responses:	
nn , mm	n and m are parameters relating to the system response. These parameters are used by optimisation programs.
! TUNE FAILURE	A stable set of coefficients could not be found.
! EMERGENCY STOP	The Emergency Stop has been activated.
! TRACKING ABORT	Controller has aborted due to a Tracking error.
! MOTOR STALLED	Controller is aborted due to stalled motor or encoder loss.
! USER ABORT	Controller is aborted due to a user command.
! HARD LIMIT	Tune attempted when already on hard limit.

Example:	
1TUNE	Tune coefficients on axis 1 controller.

<b>UL</b>	<b>SET UPPER SOFT LIMIT POSITION</b>
-----------	--------------------------------------

This command will set the Upper Soft Limit Position to the value given in the argument. Subsequent moves by the Move Absolute (**MA**) or Move Relative (**MR**) and manual Jog moves will not be allowed above this Upper Limit if the Soft Limits are enabled.

Syntax	Units	Range	to	Initial Value
<ad>ULnnn	Steps	-2147483647	2147483647 ( $\pm 2^{32}$ ).	2147483647 ( $2^{32}$ )

Condition Requirements	Notes:
Idle or Constant velocity	Value retained on power-up.

Responses	
OK	Command has been accepted.
! LIMITS DISABLED	A warning that the soft limits are currently disabled.
! LIMIT CONFLICT	Attempting to set upper limit below or equal to lower limit

Example:	
1UL8000	Set the axis 1 Upper Soft Limit Position to 8000.

<b>UP</b>	<b>UNDEFINE PROFILE</b>
-----------	-------------------------

This command will undefine or cancel a Profile definition. This will then free the memory used by the Profile for use in other sequences or profile definitions.

Syntax	Units	Range	to	Initial Value
<ad>UP	N/A	N/A		N/A

<b>Condition Requirements</b>	<b>Notes:</b>
Idle.	

<b>Responses:</b>	
OK	Command has been accepted.

**Example:**  
1UP                      Delete Profile from axis 1 controller.

<b>US</b>	<b>UNDEFINE SEQUENCE</b>
-----------	--------------------------

This command will undefine or cancel a sequence definition. This will then free the memory used by the sequence for use in other sequences or profile definitions.

Syntax	Units	Range	to	Initial Value
<ad>USn	Seq. No.	0	7	N/A

<b>Condition Requirements</b>	<b>Notes:</b>
Idle.	If the sequence specified has been set to auto-execute, then the auto-execute flag will be reset.

<b>Responses:</b>	
OK	Command has been accepted.
! OUT OF RANGE	Argument (sequence number) is out of valid range.

**Example:**  
1US6                      Delete sequence 6 from axis 1 controller.



**VERIFY PROFILE**

Syntax	Units	Range	to	Initial Value
<ad>VP	N/A	N/A	N/A	N/A

**Notes:**  
Arguments with a value of zero are not shown.

The command will either respond with the axis address identifier (if address prefix enabled) followed by each line of the Profile, or an error message:

Profile has not been defined yet.

1MR2000	First definition of Profile.
1MR7000	First move.
1MR1000	Next move.
1MR0	"
1EP	End of Profile definition.

```
1:
MR 2000
MR 7000
MR 1000
MR
OK
```

### VERIFY SEQUENCE

This command will list a previously defined Sequence.

<b>Syntax</b>	<b>Units</b>	<b>Range</b>	<b>to</b>	<b>Initial Value</b>
<ad>VS <sub>n</sub>	Seq. No.	0	7	N/A

### Condition Requirements

None.

**Notes:**

Arguments with a value of zero are not shown.

## Responses

The command will either respond with the axis address identifier (if address prefix enabled) followed by each line of the sequence, or an error message:

! OUT OF RANGE	Argument (sequence number) is out of valid range.
! NO SEQUENCE	Sequence specified has not been defined yet.

**Example:**

A controller that had previously been programmed with:

IDS2	Start definition of sequence 2.
1MA2000	First move (absolute).
1MR7000	Next move (relative).
1DE1000	Delay for 1 second.
1MA0	Next move (return to start position).
1XS2	Execute sequence 2 (loop to start of this sequence).
IES	End of sequence definition.

The command `1VS2` would give:

```
1:
MA 2000
MR 7000
DE 1000
MA
XS 2
OK
```

<b>WA</b>	<b>WAIT FOR INPUT PORT CONDITION</b>
-----------	--------------------------------------

This command will examine the read port inputs and compare them with the specified bit pattern argument. It will wait until the inputs are equal to the specified bit pattern before issuing its 'OK' response and moving on to the next command.

The bit pattern is specified as a four digit binary number of either 0, 1 or 2 characters starting with read port 4, through to 1. A 0 defines that the input must be high (+24V), a 1 defines that the input must be low (0V or open-circuit) and a 2 defines that the input is not relevant or 'don't care'. If less than four digits are specified in the argument, then the preceding ones are assumed as low (0).

Syntax	Units	Range	Initial Value
<ad>WAbbbb	Bit pattern	4 digits of 0, 1 or 2	N/A

Condition Requirements	Notes:
None.	

Responses	
OK	Command has been accepted.
! WA SYNTAX	Invalid argument i.e. bit specified was not 0, 1 or 2 OR the Number of bits was greater than 4.

**Example:**  
1WA2210 Will wait until the following condition is on the read input port before continuing:

PORT:	4	3	2	1
STATE:	(Ignored)	(Ignored)	Low	High

<b>WE</b>	<b>WAIT FOR END</b>
-----------	---------------------

This command will wait for the end of a move or delay. It will wait until any current move or delay has finished and detects the return to the *idle* state. The 'OK' response will not be issued until the move or delay has been completed. Therefore **WE** can be used to execute I/O commands after a move is complete.

Syntax	Units	Range	to	Initial Value
<ad>WE	N/A	N/A		N/A

Condition Requirements	Notes:
None.	

Response:	
OK	Command has been completed.

**Examples:**

1MR4000	Move 4000 steps positive.
1WE	Wait for End of above move
1WP2220	Turn LED on (write port 1) when move has finished.
1DE1000	Delay for 1 second.
1WE	Wait for End of Delay
1WP2221	Turn LED off (write port 1).

<b>WI</b>	<b>SET WINDOW</b>
-----------	-------------------

This command will set the window for end of move checking. At the end of a move, when the motor comes within the **WI** range of this final target, the **SE** (settling time) counter counts down. When the settling time reaches zero the controller will either accept the next command or go to the *idle* condition.

If the motor overshoots the window before to the settling time reaches zero, the settling time counter is reset and started again.

Syntax	Units	Range	to	Initial Value
<ad>WInnn	Steps	0	2147483647 (2 <sup>32</sup> )	4

Condition Requirements	Notes:
None.	Value retained on power-up.

Responses	
OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

**Example:**  
1WI2                      Set the Window for axis 1 to 2 steps.

<b>WP</b>	<b>WRITE TO OUTPUT PORT</b>
-----------	-----------------------------

Write to output port. The PM300 controller has four user output ports, known as write ports 1 to 4. This command will set the write port outputs to a state defined by the specified bit pattern argument. The bit pattern is specified as a four digit binary number. The digits will be either 0, 1 or 2 characters starting with write port 4 through to 1

Format:        Four digit binary string  
                 consisting of 0s, 1s or 2s.  
                 0 = *On* +24V (depending on the voltage of Write Port V<sub>source</sub>)  
                 1 = *Off* 0V or open-circuit  
                 2 = *Don't change*

Syntax	Units	Range	Initial Value
<ad>WPbbbb	Bit pattern	4 digits of 0, 1 or 2	N/A

Condition Requirements	Notes:
None.	Initial state on power-up: all 1 = <i>Off</i> The last <i>write</i> is shown on the <b>QA</b> page.

Responses:	
OK	Command has been accepted.
! WP SYNTAX	Invalid argument i.e. bit specified was either not 0, 1 or 2 or the Number of bits was greater than four.

**Example:**  
If a PM300 on axis 1 currently has the following states on its output write ports:

PORT:	4	3	2	1
STATE:	on	on	off	off

1WP1200    Will set the outputs to:

PORT:	4	3	2	1
STATE:	off	on	on	on
	1	2 (unchanged)	0	0

<b>XP</b>	<b>EXECUTE PROFILE</b>
-----------	------------------------

This command will execute the defined Profile. The move occurs at a rate, defined in milliseconds, for each **MR** segment to be completed.

<b>Syntax</b>	<b>Units</b>	<b>Range</b>	<b>to</b>	<b>Initial Value</b>
<ad>XPnnn	milliseconds	1	65535	N/A

<b>Condition Requirements</b>	<b>Notes:</b>
Idle.	

<b>Responses:</b>	
OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.
! NO PROFILE	Profile has not been defined yet.
! EMERGENCY STOP	The Emergency Stop has been activated.
! MOTOR STALLED	Controller is aborted due to stalled motor or encoder loss.
! TRACKING ABORT	Controller has aborted due to a Tracking error.
! USER ABORT	Controller is aborted due to a user command.

**Example:**  
 LXP100            Axis 1, execute Profile. Each segment takes 100 mS.

<b>XS</b>	<b>EXECUTE SEQUENCE</b>
-----------	-------------------------

This command will start execution of a sequence. The argument selects which sequence is to be executed (0 to 7). The sequence must have already been defined with a Define Sequence **DS** command.

If the Execute Sequence (**XS**) command is encountered during a sequence, it will explicitly transfer control to the beginning of the sequence specified, whether it is the sequence already running or another sequence. It may therefore be used to make a loop type sequence or jump to any other sequence. Please note that it should not be considered as a subroutine. It is like a GOTO rather than a GOSUB.

A sequence execution may be stopped before completion, or if in a continuous loop, by a Control-C or ESCAPE command.

Control-C will stop any movement immediately, exit the sequence and return to idle.

ESCAPE will decelerate any move to a stop, exit the sequence and return to idle.

<b>Syntax</b>	<b>Units</b>	<b>Range</b>	<b>to</b>	<b>Initial Value</b>
<ad>XSnn	Seq. No.	0	7	N/A

<b>Condition Requirements</b>	<b>Notes:</b>
None.	

<b>Responses:</b>	
! OUT OF RANGE	Argument (sequence number) is out of valid range.
! NO SEQUENCE	Sequence specified has not been defined yet.

Other responses may be generated by commands within the sequence. At the completion of the sequence, the response to the last command is sent.

**Example:**  
 IXS1            Execute sequence 1